

Выделяется два типа разработчиков сайтов:

Backend– занимается разработкой внутренней части сайта, программирует основной функционал;

Frontend – занимается внешней частью сайта, настраивает отображение всех элементов, добавляет к ним анимацию и так далее.

Верстка – это про Frontend, в основу которого входит язык разметки HTML, базис любого сайта.

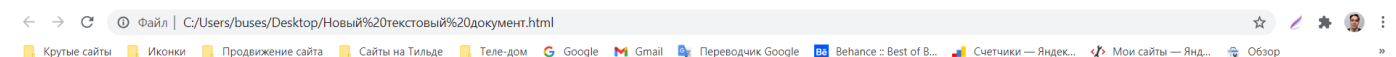
HTML позволяет показывать страницы и контент, размещенный на них, в заданном порядке. Работа с языком разметки заключается в описании тегов. Вот основные из них:

1. `<html> </html>` – главный тег, в котором содержатся другие теги;
2. `<head> </head>` – включает в себя различные SEO-элементы, код JavaScript и многое другое;
3. `<body> </body>` – внутри этих тегов находится все содержимое страницы;
4. `<h1> </h1>` – используется для обозначения заголовка первого уровня;
5. `<h2> </h2>` – используется для обозначения заголовка второго уровня, после которого следуют заголовки h3, h4, h5, h6;
6. `<p> </p>` – здесь прописывается текстовое описание, например заголовок статьи;
7. ` ` – придает тексту жирность;
8. `<i> </i>` – текст, написанный внутри этого тега, отображается курсивом;
9. ` ` – обозначает маркированный список;
10. ` ` – обозначает нумерованный список;
11. ` ` – указывает на пункты внутри списка;
12. `<a> ` – позволяет добавить к тексту ссылку на другую страницу, номер телефона, почту или иной элемент с помощью атрибута href;
13. `` – используется, чтобы добавить картинку на сайт с компьютера либо из интернета;
14. `<table> </table>` — тег для создания таблицы.

Теги работают следующим образом:

```
<h1>Привет – это мой первый сайт!</h1>
```

В таком случае на странице будет отображен заголовок h1 со стандартным шрифтом, размером и начертанием.



Привет – это мой первый сайт!

Аналогичным образом прописываются другие теги, в результате чего получается готовый сайт.

Виды верстки

Существует два вида верстки – блочная и табличная.

Табличная верстка

Первый вид верстки, с которого началась эпоха сайтов. Именно через таблицы создавались простые веб-ресурсы в далеком 2000-м году. При табличной верстке страница поделена на соседствующие ячейки, что напоминает стандартную работу с таблицами в Excel.

Минус такого подхода состоял в том, что приходилось создавать дополнительные таблицы, которые впоследствии могли остаться пустыми. Например, если требовалось разместить изображение и зафиксировать его положение, то необходимо было создать новую строку и разделить ее на несколько столбцов. Только один из них бы содержал изображение, а другие служили бы для него фиксаторами.

Таким образом, страница могла содержать большое количество пустых таблиц, из-за которых сайт становился «тяжелым». Мало того, что такой сайт долго грузится, на него еще не любят заходить поисковые роботы для индексации страниц.

Как таковая табличная верстка сейчас не используется, но без нее не обходятся при верстке электронных писем – там она, можно сказать, обязательна. Сама верстка разрабатывается с помощью тега `<table>`, который задает основные параметры таблицы – длину, ширину и прочее. Внутри тега располагаются теги `<tr>` и `<td>`, где первый необходим для создания строки, а второй – для столбца.

	<code><td></code>	<code><td></code>	<code><td></code>
<code><tr></code>	Content	Content	Content
<code><tr></code>	Content	Content	Content
<code><tr></code>	Content	Content	Content

Блочная верстка

Самый актуальный вид верстки сайтов – блочный. Он основан на теге `<div>`, с помощью которого создаются контейнеры, включающие в себя весь контент страницы или отдельного блока. Например, мы можем разделить сайт на несколько блоков: первый экран, о компании, контакты – для каждого блока будет отведен свой тег `<div>`.

Внутри тега `<div>` уже находятся другие теги, отвечающие за те или иные элементы. Вот пример небольшого блока:

```
<div>

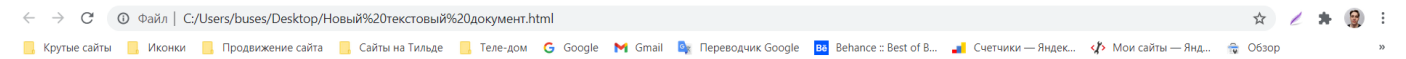
    <h1>Привет – это мой первый сайт!</h1>

    <p>Сегодня 2021 год и я сделал свой первый сайт...</p>

</div>
```

Прописав его в HTML-документе, получим следующую страницу:



Привет – это мой первый сайт!

Сегодня 2021 год и я сделал свой первый сайт...

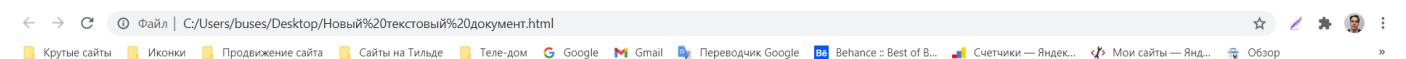


При таком подходе язык разметки HTML всегда взаимодействует с CSS-стилями, которые преобразуют обычную страницу в стильное дизайнерское решение: добавляются цвета, устанавливаются отступы для элементов, задается базовая анимация и многое другое.

Например, у нас есть тег `h1`, и мы хотим сделать его красным – для этого в стилях прописывается следующий код:

```
h1{  
  
color: red;  
  
}
```

Заголовок нашей страницы примет следующий вид:



Привет – это мой первый сайт!

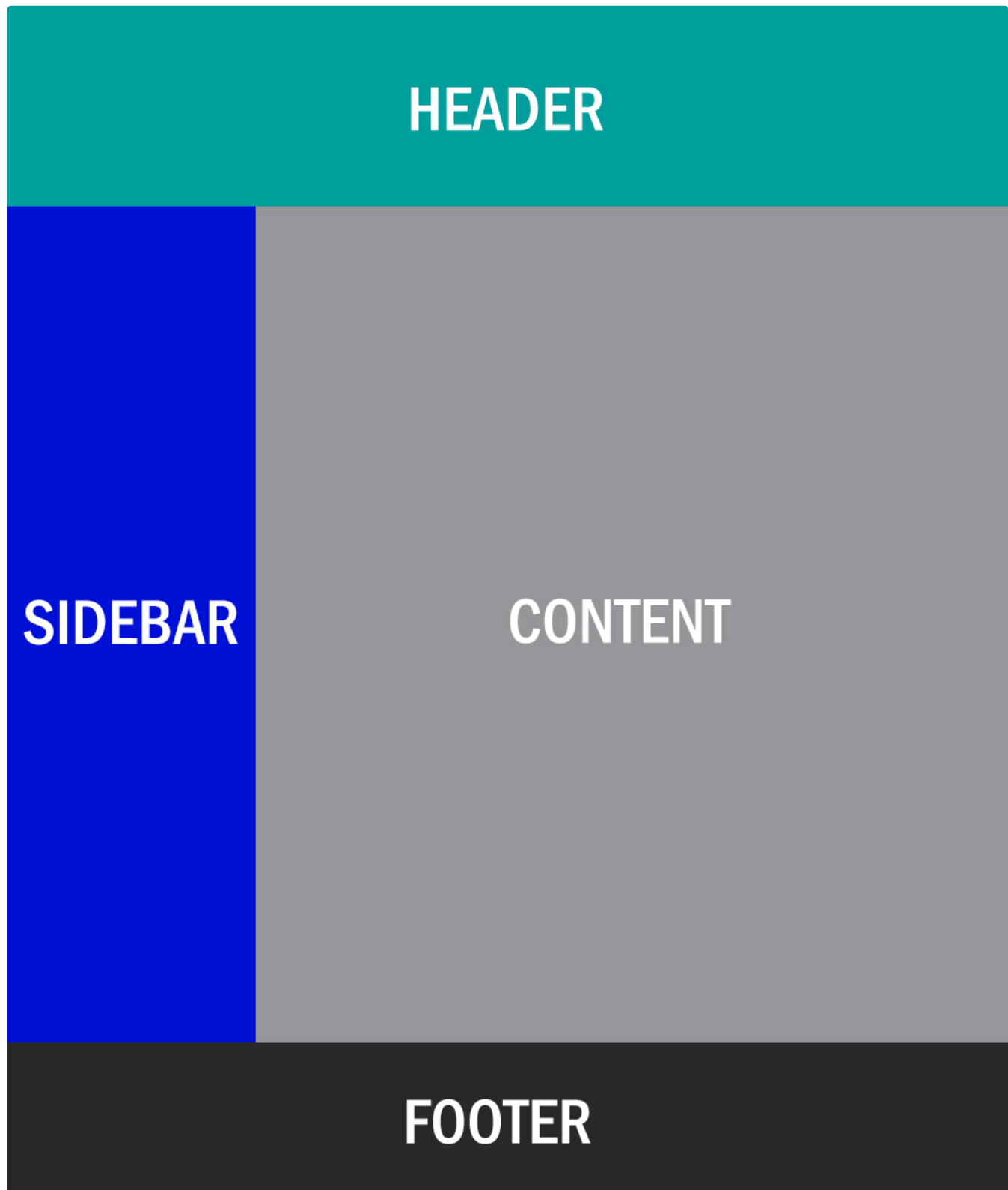
Сегодня 2021 год и я сделал свой первый сайт...



HTML и CSS обычно хранят в разных файлах – такой подход позволяет быстро вносить изменения и не путаться в больших проектах.

Также стоит сказать, что блочная верстка позволяет легко создать адаптивный сайт, что в наше время является обязательным требованием для каждого проекта. Такая разработка позволяет не только создавать сайты для телефонов и планшетов, но и обеспечивает попадание сайта в топ выдачи поисковых систем.

Вот так выглядит типичная схема блочной верстки:



Валидная верстка

Валидная верстка – это верстка, соответствующая стандарту [W3C](#) и означающая корректное отображение сайта на всех пользовательских устройствах. Такой подход обеспечивается строгим соблюдением правил построения кода, его оптимизации и минимизации. В результате это позволяет выводить сайт на более высокий уровень – его репутация в поисковиках сильно улучшается.

При валидной верстке элементы HTML-кода должны находиться строго на своих местах, в начале обязательно прописывается тег `<!DOCTYPE html>`, а также `<html>`, `<head>`, `<body>`.

Когда верстка считается правильной

Существует множество правил, которых следует придерживаться во время верстки. Нарушение некоторых из них может привести к плохой оптимизации сайта.

Рекомендуется разбивать HTML-код, CSS-стили и JavaScript-код на разные файлы.

Весь написанный код должен быть чистым и легко читаемым.

Верстка макета сайта чаще всего должна быть пиксель в пиксель, но в некоторых случаях допускаются погрешности, согласованные с заказчиком.

Для оптимизации кода рекомендуется использовать методологию БЭМ от Яндекса.

Сайт должен работать одинаково во всех браузерах.

Используйте заголовки H1-H6, а также знайте, что H1 – единственный заголовок на одной странице. Важно соблюдать это правило, иначе могут появиться проблемы с поисковыми системами.

Также для поисковиков важно, чтобы были заполнены атрибуты `<title>`, `<alt>` и `<description>`.

Не используйте «тяжелые» изображения, чтобы избежать длительной загрузки сайта.

Сайт должен быть адаптирован для мобильных устройств.

Естественно, что это не все основы «идеальной» верстки, но соблюдение даже этих правил позволит создать быстрый и устойчивый сайт.

Инструменты для верстки сайта

Верстальщик – это не простой разработчик, который работает с блокнотом и пишет в нем теги да атрибуты.

Если говорить об инструментах разработчика, то чаще всего используются:

Notepad++ – простой редактор кода;

SublimeText – наиболее используемый редактор;

Webstorm – самый мощный редактор.

Последняя программа для верстки сайтов платная, однако для обучения можно получить ее студенческую версию даром и с полным функционалом.

Как проверить верстку

После того как сайт будет сверстан, потребуется проверить его работоспособность. Базовое тестирование включает в себя проверку адаптивности на всех устройствах.

Более обширную информацию можно получить на сервисах [Jigsaw.W3](#) и [Validator.W3](#).

Если требуется узнать, соответствует ли верстка макету пиксель в пиксель, то для этого подойдет сервис [WellDoneCode](#).



устройства.



Разработка

Hello World! Гайды и обзоры для девелоперов разных мастей.

Поделитесь постом с друзьями



Комментарии

Написать комментарий

Рекомендуем

Как работать с фоновыми рисунками в CSS

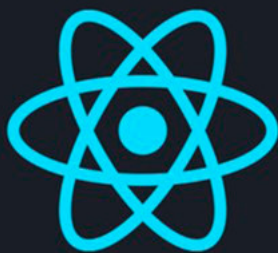
22 июня в 2022 👁 25550 💬 2



background-size: no-repeat;
background-position: center;
background-size: cover;
width: 400px;

Создаем мобильное приложение на React Native. Часть 1: Настройка окружения

21 марта в 2022 👁 12638 💬 0



Делаем мобильное приложение на React Native. Часть 1

Свой блог на базе WordPress и SvelteKit. Часть 2: Деплой в Timeweb и базовая стилизация

24 фев в 2022  1970  0





Свой блог на базе WordPress и SvelteKit. Часть 1: Запуск на локальной машине

22 фев в 2022  3011  3



Создаем блог на Svelte. Часть 11: Рефакторинг

15 фев в 2022  1450  0



Делаем собственный блог
на Svelte. Часть 1 1